

MỤC LỤC

1. Giới thiệu chung	2
1.1. Khái niệm	2
1.2. Lợi ích	2
1.3. Các vấn đề khi không quản lý phiên bản database	2
1.4. ChangeLog	2
1.5. Tạo changelog file	2
1.6. Chạy changelog	3
1.7. LiquiBase Functionality	3
2. Hướng dẫn áp dụng vào dự án cụ thể	4
2.1. Hướng dẫn sử dụng liquibase qua jenkins	4
<i>Bước 1: Tạo một project trên gitlab chứa tất cả các script và changelog được sử dụng.</i>	4
<i>Bước 2: Tạo một job trên jenkins ở dạng pipeline và thực hiện cấu hình.</i>	5
2.2. Kết quả sau khi chạy job	15

HƯỚNG DẪN SỬ DỤNG LIQUIBASE ĐỂ QUẢN LÝ PHIÊN BẢN DATABASE

1. Giới thiệu chung

1.1. *Khái niệm*

LiquiBase là một phần mềm opensource, cho phép theo dõi, quản lý và cho phép các thay đổi trên database.

Những thay đổi trên database được lưu trong một file XML và có thể được kiểm tra từ source code.

LiquiBase thực hiện các thay đổi dựa trên nội dung trong file XML để xử lý các phiên bản khác nhau của cấu trúc và dữ liệu trong database.

1.2. *Lợi ích*

LiquiBase hỗ trợ thay đổi dữ liệu một cách nhất quán

- Quản lý database ở các trạng thái khác nhau
- Lưu lại lịch sử các thay đổi
- Hỗ trợ rollback tự động
- Có khả năng tự động hóa
- Hạn chế lỗi tác động từ người dùng
- Phù hợp với nhiều loại database khác nhau như mariadb, Mysql, Oracle,...

1.3. *Các vấn đề khi không quản lý phiên bản database*

- Thay đổi cơ sở dữ liệu không nhất quán
- Không hiệu quả trong việc quản lý các thay đổi
- Tác động vào cơ sở dữ liệu có thể không được phát hiện
- Database không đồng bộ giữa các môi trường.

1.4. *ChangeLog*

Changelog là một file xml chứa tất cả các thay đổi tới database được liệt kê. Một file changelog chứa một bộ các changeset liệt kê từng thay đổi đơn lẻ trong database.

1.5. *Tạo changelog file*

Đối với một database đã tồn tại, có thể liệt kê ra danh sách dữ liệu database hiện tại.

LiquiBase giúp việc này thực hiện một cách dễ dàng bằng cách chạy lệnh đơn giản từ máy client để lấy ra tất cả các thay đổi của database

Ví dụ:

```
liquibase --driver=com.mysql.jdbc.Driver --classpath=/path/to/classes --  
changeLogFile=/path/to/db.changelog.xml --url=jdbc:mysql://hostname/database -  
-username=dbusername --password=dbpassword generateChangeLog
```

1.6. Chạy changelog

Khi chạy file changelog lần đầu tiên, LiquiBase quản lý các thay đổi đó bằng cách thêm hai bảng vào cơ sở dữ liệu.

- databasechangelog: duy trì các thay đổi cơ sở dữ liệu đã được chạy.
- databasechangeloglock: đảm bảo máy client không cố gắng sửa đổi cơ sở dữ liệu cùng một lúc.

Ví dụ:

```
liquibase --driver=com.mysql.jdbc.Driver --classpath=/path/to/classes --  
changeLogFile=/path/to/db.changelog.xml --url=jdbc:mysql://hostname/database -  
-username=dbusername --password=dbpassword migrate
```

1.7. LiquiBase Functionality

LiquiBase refactoring functionality:

- **Structural:** Columns, Tables, Views, Stored Procedures
- **Data Quality:** Columns, Tables, Views, Stored Procedures
- **Referential Integrity:** Foreign Keys, Primary Keys
- **Transformations:** Insert, Update, Delete, Tag, Stop
- **Architectual:** Indexes
- **Custom:** Custom SQL, Modify SQL, Execution

2. Hướng dẫn áp dụng vào dự án cụ thể

Đối với mỗi dự án cụ thể, việc quản lý version database sẽ phụ thuộc vào tác động của dev đối với hệ thống.

Database có thể thay đổi trong các trường hợp:

- Thay đổi cấu trúc của table, thay đổi tên column,...
- Thêm table, thêm column khi phát triển tính năng mới,...
- Đồng bộ thay đổi của database đối với các môi trường khác nhau như dev, staging, productions,...

Yêu cầu đối với việc quản lý database với liquibase:

- Khuyến nghị việc sử dụng database để đồng bộ schema của database
- Hoặc sử dụng để thêm data fetch hỗ trợ test.

Liquibase được cài trên server jenkins và chạy qua command line.

Hạn chế:

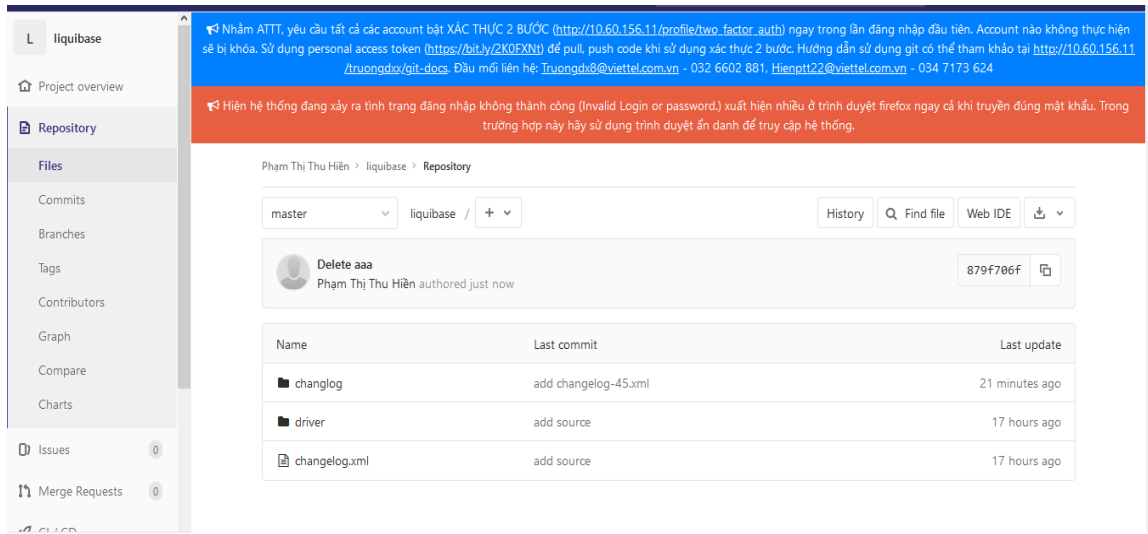
- Khi chạy qua command line sẽ liên quan đến việc lộ mật khẩu để có thể truy cập vào database.
⇒ Vì vậy khuyến nghị chạy qua jenkinsfile tức là chạy qua job jenkins ở dạng pipeline.
- Để thực hành đối với liquibase trong việc quản lý version cho database nên tạo một schema test để có thể thực hành → tránh ảnh hưởng tới db hiện tại của dự án.

2.1. Hướng dẫn sử dụng liquibase qua jenkins

Bước 1: Tạo một project trên gitlab chứa tất cả các script và changelog được sử dụng.

⇒ Không thực hiện xóa changelog hay script để tránh việc run job bị faild

Lý do: Liquibase sẽ thực hiện kiểm tra các thay đổi dựa vào bảng databasechangelog. Nếu sửa file script chạy job sẽ ở trạng thái faild or unstable



Trong đó:

- **Changelog:** là folder chứa log chạy diff và file changelog chứa tất cả thay đổi của database để chạy đồng bộ trên schema target.
- **Driver:** chứa file classpath connect tới server
- **Changelog.xml** là file chạy để chứa tất cả các file chạy thay đổi.

Bước 2: Tạo một job trên jenkins ở dạng pipeline và thực hiện cấu hình.

Có hai cách run script: run qua jenkinsfile hoặc viết trực tiếp script pipeline trên jenkins.

Tạo job: Chọn New Item → Nhập tên job → Chọn loại job (**PIPELINE**)

Cấu hình script run pipeline: có jenkinsfile đính kèm. Phần dưới đây sẽ giải thích từng stage chạy qua jenkins.

a. Cấu hình agent chạy job

```
agent { node ("master") }
```

b. Cấu hình lấy source code trên gitlab

```
stage('checkout sourcecode') {
```

```
    steps {
```

```

        git credentialsId: '63265de3-8396-40f9-803e-5cd0b694e519', url:
        'http://10.60.156.11/hienptt22/liquibase.git'
    }
}

```

Trong đó:

- **credentialsId**: ID lấy trong gitlab khi add Credential
 - **url** : URL dùng để clone source code.
- c. Cấu hình kiểm tra thay đổi và tạo file changelog để chạy migrate

```

stage('Get changelog file') {
    steps {
        withCredentials([
            usernamePassword(credentialsId: '37c1ec3c-89e4-4658-9909-
c611fe282f44', usernameVariable: 'username1', passwordVariable: 'password1'),
            usernamePassword(credentialsId: '37c1ec3c-89e4-4658-9909-
c611fe282f44', usernameVariable: 'username2', passwordVariable: 'password2')
        ]){
            sh """

            ${LIQUIBASE_HOME}/liquibase --changeLogFile=changelog.xml \
            --outputFile=changelog/changelog-${BUILD_NUMBER}.txt \
            --driver=org.mariadb.jdbc.Driver --classpath=driver/mariadb-java-client-
2.4.0.jar \
            --url="jdbc:mariadb://10.60.156.43:3306/test" --username=${username1} \

```

```

--password=${password1} \

--diffTypes=tables,indexes,views,columns,primaryKeys,data \

diff \

--referenceUrl="jdbc:mariadb://10.60.156.43:3306/test1" \

--referenceUsername=${username2} \

--referencePassword=${password2}

""

sh ""

${LIQUIBASE_HOME}/liquibase \

--changeLogFile=changelog/changelog-${BUILD_NUMBER}.xml \

--outputFile=changelog/changelog-${BUILD_NUMBER}.txt \

--driver=org.mariadb.jdbc.Driver --classpath=driver/mariadb-java-client-
2.4.0.jar \

--url="jdbc:mariadb://10.60.156.43:3306/test" \

--username=${username1} \

--password=${password1} \

--diffTypes=tables,indexes,views,columns,primaryKeys \

diffChangeLog \

--referenceUrl="jdbc:mariadb://10.60.156.43:3306/test1" \

--referenceUsername=${username2} --referencePassword=${password2}

""

```

```

    }
}
}

```

Trong đó:

1. withCredentials([

```

    usernamePassword(credentialsId: '37c1ec3c-89e4-4658-9909-
c611fe282f44', usernameVariable: 'username1', passwordVariable: 'password1'),

```

```

    usernamePassword(credentialsId: '37c1ec3c-89e4-4658-9909-
c611fe282f44', usernameVariable: 'username2', passwordVariable: 'password2')

```

- **credentialsId**: Id của Credential add trong jenkins → Đây là user/pass connect tới database
- **usernameVariable**: username đăng nhập database username1/username2 là biến tự đặt tên. Trong trường hợp này sử dụng 1 cặp credential dùng để kết nối tới 2 database test và staging.
- **passwordVariable** tương tự usernameVariable

2. diff → kiểm tra sự khác nhau giữa hai database. Không tạo ra file changelog.

```
sh ""
```

```

${LIQUIBASE_HOME}/liquibase \
--changeLogFile=changelog.xml \
--outputFile=changelog/changelog-${BUILD_NUMBER}.txt \
--driver=org.mariadb.jdbc.Driver \
--classpath=driver/mariadb-java-client-2.4.0.jar \
--url="jdbc:mariadb://10.60.156.43:3306/test" \

```



```
--username=${username1} \

--password=${password1} \

--diffTypes=tables,indexes,views,columns,primaryKeys,data \

diff \

--referenceUrl="jdbc:mariadb://10.60.156.43:3306/test1" \

--referenceUsername=${username2} \

--referencePassword=${password2}
```

""""

- **\${LIQUIBASE_HOME}**: là đường dẫn trỏ đến file thực thi liquibase
- **changeLogFile** : cấu hình đường dẫn tới file changelog master → file cấu hình chung hoặc có thể trỏ đường dẫn tới một file cụ thể
- **outputFile**: cấu hình đường dẫn tới file lưu kết quả của lệnh. Lưu ý: tạo file với tên khác nhau. Log này có thể bị ghi đè khi chạy lần sau
- **driver, classpath**: cấu hình driver, classpath connect tới các loại database khác nhau. Cụ thể như sau:

Loại Database	Driver	Classpath
Mariadb	org.mariadb.jdbc.Driver	driver/mariadb-java-client-2.4.0.jar
Oracle	oracle.jdbc.OracleDriver	driver/ojdbc8.jar
Mysql	com.mysql.cj.jdbc.Driver	driver/mysql-connector-java-8.0.17.jar
Postgresql	org.postgresql.Driver	driver/postgresql-42.2.5.jre6.jar
Sqllite	org.sqlite.JDBC	driver/sqlite-jdbc-3.23.1.jar

- url: Url connect tới database1
- User/pass: user/pass connect tới database1
- diffTypes: type để so sánh giữa hai database

- `referenceUrl`: Url connect tới database muốn so sánh (database ở môi trường khác) với database đang sử dụng
 - `referenceUsername/ referencePassword`: user/pass connect tới database2
3. **diffChangeLog** → Tạo file **changelog** chứa sự thay đổi giữa hai database khác nhau. Tương tự như diff nhưng **diffTypes** trong **diffChangeLog** không được sử dụng type là data.
- **changeLogFile**: đường dẫn trỏ đến file chứa các thay đổi. file changelog.xml sẽ được tạo ra theo tên đặt trong lệnh. Lệnh sẽ chạy lỗi khi file đã tồn tại.

sh ""

```

${LIQUIBASE_HOME}/liquibase

--changeLogFile=changelog/changelog-${BUILD_NUMBER}.xml \

--outputFile=changelog/changelog-${BUILD_NUMBER}.txt \

--driver=org.mariadb.jdbc.Driver \

--classpath=driver/mariadb-java-client-2.4.0.jar \

--url="jdbc:mariadb://10.60.156.43:3306/test" \

--username=${username1} \

--password=${password1} \

--diffTypes=tables,indexes,views,columns,primaryKeys \

diffChangeLog \

--referenceUrl="jdbc:mariadb://10.60.156.43:3306/test1" \

--referenceUsername=${username2} \

--referencePassword=${password2} \

```

""

- d. Cấu hình để người dùng review thay đổi và cho phép thực thi file thay đổi đã tạo ra

```
stage ('Wait for maintainer confirm change log file'){
  steps {
    script{
      echo "Check content file changelog-${BUILD_NUMBER}.xml"
      sh 'cat changlog/changelog-${BUILD_NUMBER}.xml'
      timeout(time: 1, unit: 'HOURS') {
        deployInput = input(
          message: 'Pause for wait maintainer selection', ok: "Execute",
        parameters: [
          choice(choices: ['Deploy', 'Abort'], description: 'Deploy version database
with changelog-${BUILD_NUMBER}.xml?', name: 'DEPLOY_CHOICES')
        ])
      }
      if (deployInput == 'Deploy') {
        withCredentials([
          usernamePassword(credentialsId: '37c1ec3c-89e4-4658-9909-
c611fe282f44', usernameVariable: 'username1', passwordVariable: 'password1'
        )]){
          sh """

          ${LIQUIBASE_HOME}/liquibase \
          --changeLogFile=changelog/changelog-${BUILD_NUMBER}.xml \
          --outputFile=changelog/changelog-${BUILD_NUMBER}.txt \
```


File được tạo ra sẽ ở dạng pattern **changelog-\${BUILD_NUMBER}.xml** với **BUILD_NUMBER** là bản build hiện tại.

File được tạo ra ví dụ như sau:

```
<?xml version="1.1" encoding="UTF-8" standalone="no"?>

<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
xmlns:pro="http://www.liquibase.org/xml/ns/pro"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog-ext
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-ext.xsd
http://www.liquibase.org/xml/ns/pro http://www.liquibase.org/xml/ns/pro/liquibase-pro-
3.8.xsd http://www.liquibase.org/xml/ns/dbchangelog
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.8.xsd">

  <changeSet author="app (generated)" id="1579222511391-1">

    <addColumn tableName="tbl_book">

      <column name="description" type="VARCHAR(255 BYTE)"/>

    </addColumn>

  </changeSet>

  <changeSet author="app (generated)" id="1579222511391-2">

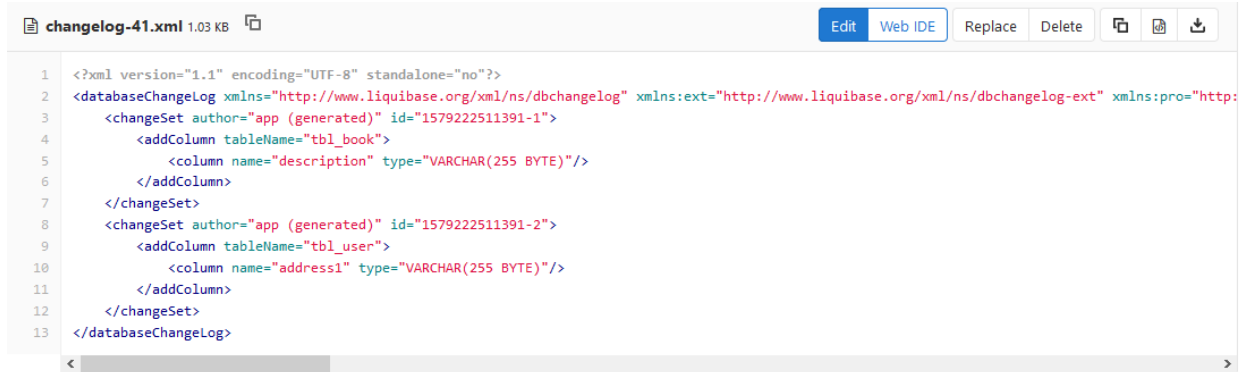
    <addColumn tableName="tbl_user">

      <column name="address1" type="VARCHAR(255 BYTE)"/>

    </addColumn>

  </changeSet>

</databaseChangeLog>
```



```
1 <?xml version="1.1" encoding="UTF-8" standalone="no"?>
2 <databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog" xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext" xmlns:pro="http://www.liquibase.org/xml/ns/pro" xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-4.2.xsd" >
3   <changeSet author="app (generated)" id="1579222511391-1">
4     <addColumn tableName="tbl_book">
5       <column name="description" type="VARCHAR(255 BYTE)"/>
6     </addColumn>
7   </changeSet>
8   <changeSet author="app (generated)" id="1579222511391-2">
9     <addColumn tableName="tbl_user">
10      <column name="address1" type="VARCHAR(255 BYTE)"/>
11    </addColumn>
12  </changeSet>
13 </databaseChangeLog>
```

- Chạy lệnh migrate và đánh tag cho sự thay đổi của database như lệnh ở phần
bôi đỏ.

e. Commit tất cả thay đổi tới gitlab để quản lý

```
stage ("Commit data to gitlab"){
```

```
  steps {
```

```
    script {
```

```
      withCredentials([usernamePassword(credentialsId: '63265de3-8396-40f9-803e-5cd0b694e519', usernameVariable: 'username',
```

```
                                passwordVariable: 'password'))){
```

```
        sh """
```

```
          git config --global user.name "${username}"
```

```
          git config --global user.email "${username}.viettel.com.vn"
```

```
          git config --global push.default matching
```

```
          git add .
```

```
          git commit -m "add changelog-${BUILD_NUMBER}.xml"
```

```
          git push 'http://${username}:${password}@10.60.156.11/hienptt22/liquibase.git'
```

""""

}

}

}

}

Lưu ý: sửa đường dẫn link git trở đúng project đã tạo trên jenkins.

2.2. Kết quả sau khi chạy job

Sau khi chạy script trong database sẽ lưu lại thay đổi trong databaselog

```
5 08:54:12 | 3 | EXECUTED | NULL | sqlFile | NULL
| insert book | hienptt22 | flyway/changelog-1.0.xml | 2020-01-1
5 08:55:17 | 3.8.5 | 4 | EXECUTED | NULL | sqlFile | NULL
| insert book1 | hienptt22 | flyway/changelog-1.0.xml | 2020-01-1
5 08:55:17 | 3.8.5 | 5 | EXECUTED | NULL | sqlFile | NULL
| insert data user | hienptt22 | changelog-1.1.xml | 2020-01-1
5 16:25:07 | 3.8.5 | 6 | EXECUTED | NULL | sqlFile | liquibase_vers
| insert a data user | hienptt22 | changelog-1.1.xml | 2020-01-1
5 16:27:36 | 3.5.3 | 7 | EXECUTED | NULL | sqlFile | liquibase_vers
| raw | includeAll | /u02/jenkins/workspace/liquibase_versioning_db/sql/V1_test_insert.sql | 2020-01-1
6 09:07:40 | 3.5.3 | 8 | EXECUTED | NULL | sql | NULL
| raw | includeAll | /u02/jenkins/workspace/liquibase_versioning_db/sql/V2_Add_column_table_user.sql | 2020-01-1
6 09:07:40 | 3.5.3 | 9 | EXECUTED | NULL | sql | v1.0
| 1579166216330-1 | app (generated) | changlog/changelog-18.xml | 2020-01-1
6 16:23:18 | 3.8.5 | 10 | EXECUTED | 8:f57ffc44e54da1bbb7f38984b28fc3c3 | addColumn tableName=tbl_user | v1.27
| 1579167983789-1 | app (generated) | changlog/changelog-26.xml | 2020-01-1
6 16:46:33 | 3.8.5 | 12 | EXECUTED | 8:f57ffc44e54da1bbb7f38984b28fc3c3 | addColumn tableName=tbl_user | v1.26
| 1579223071643-1 | app (generated) | changlog/changelog-43.xml | 2020-01-1
7 08:04:40 | 3.8.5 | 13 | EXECUTED | 8:0f115bc563459780f79b6f6f57848f22 | addColumn tableName=tbl_book | NULL
| 1579223071643-2 | app (generated) | changlog/changelog-43.xml | 2020-01-1
7 08:04:40 | 3.8.5 | 14 | EXECUTED | 8:f57ffc44e54da1bbb7f38984b28fc3c3 | addColumn tableName=tbl_user | v1.48
| 1579223071643-2 | app (generated) | changlog/changelog-43.xml | 2020-01-1
```